In the Claims

The status of claims in the case is as follows:

1   1.   [Previously presented]   A method for control and
2   management of communication traffic, comprising the steps
3   of:

4   expressing access rules as filters referencing system
5   kernel data;

6   for outbound processing, determining source application
7   indicia;

8   for inbound packet processing, executing a look-ahead
9   function to determine target application indicia; said
10   look-ahead function being executed within a protocol
11   stack including an IP layer, a transport layer, a
12   sockets layer, and an application layer and which, for
13   said inbound packet, said IP layer provides to said
14   transport layer said inbound packet, marked as non-
15   deliverable, and receives back from said transport
16   layer indicia, provided to said transport layer by said

17   sockets layer, identifying the application layer

18   application to which said packet would have been

19   delivered; and

20   responsive to said source or target application

21   indicia, executing filter processing; said filter

22   processing including constructing and evaluating

23   logical expressions of arbitrary length, and

24   selectively using a set of logical operators,

25   alternative filter selector fields, and value set.

1   2.   [Currently amended] The method of claim 1, <u>wherein</u>

2   <u>said protocol stack is a TCP/IP protocol stack, and</u> further

3   comprising the steps of executing said determining and

4   executing steps within a kernel filtering function upon

5   encountering a filter selector field referencing kernel data

6   not included in said packet.

1   3.   [Currently amended] The method of claim 1, <u>wherein</u>

2   <u>said protocol stack is a TCP/IP protocol stack, and</u> said

3   filter processing including the steps of:

4   determining a task or thread identifier;

5   based on said task or thread identifier, determining a

END920010019US1   3 of 39   S/N 09/919,185

6   process or job identifier; and

7   based on said process or job identifier, determining
8   job or process attributes for filter processing.

1   4.   [Currently amended] The method of claim 1, <u>wherein</u>
2   <u>said protocol stack is a TCP/IP protocol stack, and</u> said
3   filter processing including the steps of:

4   determining a user identifier; and

5   based on said user identifier, determining user
6   attributes for filter processing.

1   5.   [Original] The method of claim 3, further comprising
2   the step of determining from said task identifier a work
3   control block containing said process or job identifier.

1   6.   [Canceled]
2   7.   [Canceled]

1   8.   [Currently amended] The method of claim 1, <u>wherein</u>
2   <u>said protocol stack is a TCP/IP protocol stack, and</u> further

END920010019US1   4 of 39   S/N 09/919,185

3   comprising the steps of:

4   delivering to said filters infrastructure access rules

5   for defining security context.

1   9.   [Original] The method of claim 8, said infrastructure

2   including logging, auditing, and filter rule load controls.

1   10.   [Previously presented] A method for control and

2   management of aspects of communication traffic within

3   filtering, comprising the steps of:

4   receiving IP packet data into a TCP/IP protocol stack

5   executing within a system kernel;

6   for an inbound IP packet, executing a look-ahead

7   function within a protocol stack including an IP layer,

8   a transport layer, a sockets layer, and an application

9   layer and which, for said IP inbound packet, said IP

10   layer provides to said transport layer said inbound IP

11   packet, marked as non-deliverable, and receives back

12   from said transport layer indicia, provided to said

13   transport layer by said sockets layer, identifying the

14   application layer application to which said packet

15   would have been delivered; and

END920010019US1   5 of 39   S/N 09/919,185

16   executing filtering code within said system kernel with

17   respect to non-IP packet data accessed within said

18   system kernel outside of said TCP/IP protocol stack;

19   said filtering code constructing and evaluating logical

20   expressions of arbitrary length, and selectively using

21   a set of logical operators, alternative filter selector

22   fields, and value set.

1   11. [Original] The method of claim 10, said non-IP packet

2   data including context data regarding said IP packet.

1   12. [Original] The method of claim 10, said non-IP packet

2   data including data specific to a task generating said non-

3   IP packet data.

1   13. [Original] The method of claim 10, said non-IP packet

2   data including data specific to a task that will receive

3   said IP packet.

1   14. [Original] The method of claim 11, said context data

2   including packet arrival interface indicia.

      15. [Canceled]

      16. [Canceled]

      17. [Canceled]

END920010019US1   6 of 39   S/N 09/919,185

1   18.   [Previously presented] A method for centralizing

2   system-wide communication management and control within

3   filter rules, comprising the steps of:

4   providing filter statements syntax for accepting

5   parameters in the form of a selector, each selector

6   specifying selector field, operator, and a set of

7   values;

8   for an inbound packet, executing a look-ahead function

9   within a protocol stack including an IP layer, a

10   transport layer, a sockets layer, and an application

11   layer and which, for said inbound packet, said IP layer

12   provides to said transport layer said inbound packet,

13   marked as non-deliverable, and receives back from said

14   transport layer indicia, provided to said transport

15   layer by said sockets layer, identifying the

16   application layer application to which said packet

17   would have been delivered by said sockets layer;

18   said selector referencing data that does not exist in

19   IP packets;

20   processing said filter statements, including

21   constructing and evaluating logical expressions of

END920010019US1   7 of 39   S/N 09/919,185

22   arbitrary length, and selectively using a set of

23   logical operators, alternative filter selector fields,

24   and value set.

1   19.   [Currently amended]   The method of claim 18,   <u>wherein</u>

2   <u>said protocol stack is a TCP/IP protocol stack, and</u> said

3   parameters selectively including userid, user profile, user

4   class, user group, user group authority, user special

5   authority, job name, process name, job group, job class, job

6   priority, other job or process attributes, and date & time.

1   20.   [Currently amended]   The method of claim 18,   <u>wherein</u>

2   <u>said protocol stack is a TCP/IP protocol stack, and</u> said

3   filters statements being provided within a user interface to

4   said system.

1   21.   [Currently amended]   The method of claim 18,   <u>wherein</u>

2   <u>said protocol stack is a TCP/IP protocol stack, and</u> further

3   comprising the steps of:

4   establishing a tunnel between two IP address limiting

5   traffic to applications bound to ports at each end of

6   said tunnel;

7   said filtering code accessing filtering attributes

END920010019US1   8 of 39   S/N 09/919,185

8   further limiting traffic selectively to job indicia;

9   and

10   operating said filtering code within a kernel filtering

11   function upon encountering a filter selector field

12   referencing kernel data not included in said traffic.

1   22.   [Currently amended] A method for traversing a portion

2   only of a protocol stack to disallow selective IP packet

3   traffic, comprising the steps of:

4   receiving a packet in the kernel of the operating

5   system of a first node from an application, said kernel

6   including a filter processor; said filter processor for

7   constructing and evaluating logical expressions of

8   arbitrary length, said logical expressions selectively

9   including a set of logical operators, alternative

10   filter selector fields, and value set;

11   for inbound packet processing to a first node from a

12   second node, executing a look-ahead function in the

13   system kernel of said first node to determine a target

14   application; said system kernel including a TCP/IP

15   protocol stack including an IP layer, a transport

16   layer, a sockets layer, and an application layer and

END920010019US1   9 of 39   S/N 09/919,185

17   which, for said inbound packet, said IP layer provides

18   to said transport layer said inbound packet, marked as

19   non-deliverable, and receives back from said transport

20   layer indicia identifying the application layer

21   application to which said packet would have been

22   delivered;

23   for both said inbound packet processing, and for

24   outbound packet processing from said first node to said

25   second node, executing within said kernel the steps of

26   processing said packet by determining a task ID;

27   responsive to said task ID, determining a

28   corresponding work control block;

29   determining a user ID, process or job identifier

30   from said work control block;

31   from the user ID, process or job identifier

32   selectively determining attributes for said user

33   process or job; and

34   passing said attributes to said filter processor

35   for managing and controlling communication

END920010019US1   10 of 39   S/N 09/919,185

36   traffic.

1   23.   [Previously presented]   A method for expressing access
2   rules as filters, comprising the steps of:

3   providing a filter statements syntax for accepting
4   parameters in the form of a selector, each selector
5   specifying selector field, operator, and a set of
6   values; and

7   said selector referencing data that does not exist in
8   IP packets for controlling access to an application;

9   for an inbound IP packet, executing a look-ahead
10   function within a protocol stack including an IP layer,
11   a transport layer, a sockets layer, and an application
12   layer and which, for said IP inbound packet, said IP
13   layer provides to said transport layer said inbound IP
14   packet, marked as non-deliverable, and receives back
15   from said transport layer indicia, provided to said
16   transport layer by said sockets layer, identifying the
17   application layer application to which said packet
18   would have been delivered; and

19   processing said filter statements by constructing and

END920010019US1   11 of 39   S/N 09/919,185

20   evaluating logical expressions of arbitrary length,

21   said logical expressions selectively including a set of

22   logical operators, alternative filter selector fields,

23   and value set referencing said application layer

24   application.

1   24.   [Previously presented]   A method for managing and

2   controlling communication traffic by centralizing access

3   rules in filters executing within and referencing data

4   available in system kernels, comprising the steps for

5   outbound packet processing from a first node to a second

6   node of:

7   receiving said packet in the kernel of the operating

8   system of said first node from an application or

9   process at said first node;

10   processing said packet by determining a task ID;

11   responsive to said task ID, determining a corresponding

12   work control block;

13   responsive to said work control block, determining a

14   process or job identifier;

15   responsive to said process or job identifier,

16   determining job or process attributes; and

17   executing said filters by constructing and evaluating

18   logical expressions of arbitrary length, said logical

19   expressions selectively including a set of logical

20   operators, alternative filter selector fields, and

21   value set.

1   25. [Previously presented] The method of claim 24, further

2   comprising the steps for inbound packet processing from said

3   second node to said first node of:

4   initially operating said kernel at said first node to

5   determine a target application for said packet at said

6   first node by executing a look-ahead function within a

7   protocol stack including an IP layer, a transport

8   layer, a sockets layer, and an application layer and

9   which, for said inbound packet, said IP layer provides

10   to said transport layer said inbound packet, marked as

11   non-deliverable, and receives back from said transport

12   layer indicia, provided to said transport layer by said

13   sockets layer, identifying the application layer

14   application to which said packet would have been

15   delivered;.

END920010019US1   13 of 39   S/N 09/919,185

26.   [Canceled]

27.   [Canceled]

28.   [Canceled]

1   29.   [Currently amended A method for managing and

2   controlling communication traffic by centralizing the access

3   rules, comprising the steps for outbound packet processing

4   from a first node to a second node of:

5   receiving said packet in the kernel of the operating

6   system of said first node from an application or

7   process at said first node, said kernel including a

8   filter processor for constructing and evaluating

9   logical expressions of arbitrary length, said logical

10   expressions selectively including a set of logical

11   operators, alternative filter selector fields, and

12   value set;

13   processing said packet within a TCP/IP stack;

14   by determining a task ID;

15   responsive to said task ID, determining a

16   corresponding work control block;

END920010019US1   14 of 39   S/N 09/919,185